Deep Learning - Week 9

1. What is the disadvantage of using Hierarchical Softmax?

   (a) It requires more memory to store the binary tree

   (b) It is slower than computing the softmax function directly

   (c) It is less accurate than computing the softmax function directly

   (d) It is more prone to overfitting than computing the softmax function directly

   Correct Answer: (c)
   Solution: The primary drawback is that the hierarchical softmax approximation can lead to less accurate probability estimates compared to the full softmax. This is because the binary tree structure imposes a fixed dependency on the vocabulary, which can negatively impact the quality of the learned representations. Therefore, the correct answer is:

2. Consider the following corpus: "AI driven user experience optimization. Perception of AI decision making speed. Intelligent interface adaptation system. AI system engineering for enhanced processing efficiency". What is the size of the vocabulary of the above corpus?

   (a) 18

   (b) 20

   (c) 22

   (d) 19

   Correct Answer: (d)
   Solution: There are 19 distinct words: [ai, driven, user, experience, optimization, perception, of, decision, making, speed, intelligent, interface, adaptation, system, engineering, for, enhanced, processing, efficiency] Therefore, the size of the vocabulary is 19.

3. We add incorrect pairs into our corpus to maximize the probability of words that occur in the same context and minimize the probability of words that occur in different contexts. This technique is called:

   (a) Negative sampling

   (b) Hierarchical softmax

   (c) Contrastive estimation

   (d) Glove representations

   Correct Answer: (a)
   Solution: Negative sampling is a technique used in word2vec and other word embedding models where:

For each positive example (words that actually occur together in context) We generate several negative examples (incorrect word pairs that don't occur together) The model is trained to:

Maximize probability for words that occur together in real contexts Minimize probability for the randomly sampled negative pairs

This helps the model learn meaningful word representations more efficiently

4. Let X be the co-occurrence matrix such that the (i, j)-th entry of X captures the PMI between the i-th and j-th word in the corpus. Every row of X corresponds to the representation of the i-th word in the corpus. Suppose each row of X is normalized (i.e., the L_2 norm of each row is 1) then the (i, j)-th entry of XX^T captures the:

   (a) PMI between word i and word j

   (b) Euclidean distance between word i and word j

   (c) Probability that word i

   (d) Cosine similarity between word i

   Correct Answer: (d)
   Solution:

   Since each row of X is normalized (i.e., L_2-norm is 1), the dot product of two normalized vectors corresponds to the cosine similarity between them.

   The matrix XX^T computes the dot products between the rows of X, which represents the similarity between the word vectors (in terms of PMI). Since the rows are normalized, the dot product gives the cosine similarity between the word representations.

   Thus, the (i, j)-th entry of XX^T captures the **cosine similarity** between word i and word j.

   The correct answer is: **cosine similarity between word** i.

5. Suppose that we use the continuous bag of words (CBOW) model to find vector representations of words. Suppose further that we use a context window of size 3 (that is, given the 3 context words, predict the target word $P(w_t|(w_i, w_j, w_k))$). The size of word vectors (vector representation of words) is chosen to be 100 and the vocabulary contains 20,000 words. The input to the network is the one-hot encoding (also called 1-of-V encoding) of word(s). How many parameters (weights), excluding bias, are there in W_{word}? Enter the answer in thousands. For example, if your answer is 50,000, then just enter 50.

   Correct Answer: 2000

   Solution: In the CBOW model, we have:

   - Vocabulary size = 20,000 words
   - Embedding size (dimension of word vectors) = 100

The weight matrix W_{word} has dimensions $100 \times 20,000$, where:

- The number of rows corresponds to the embedding size (100),
- The number of columns corresponds to the vocabulary size (20,000).

Thus, the total number of parameters (weights) is:

$$100 \times 20,000 = 2,000,000 \text{ parameters}$$

Since the question asks for the answer in thousands, the answer is:

$$\boxed{2000}$$

6. You are given the one hot representation of two words below:
GEMINI= $[1, 0, 0, 0, 1]$, CLAUDE= $[0, 0, 0, 1, 0]$
What is the Euclidean distance between CAR and BUS?

Correct Answer: range(1.7,1.74)

Solution:

The Euclidean distance between two vectors $\mathbf{A}$ and $\mathbf{B}$ is given by the formula:

$$d(\mathbf{A}, \mathbf{B}) = \sqrt{\sum_{i=1}^{n} (A_i - B_i)^2}$$

Where:

- A_i and B_i are the components of vectors $\mathbf{A}$ and $\mathbf{B}$, respectively.
- n is the number of dimensions (in this case, 5).

We are given:

$$\mathbf{A} = [1, 0, 0, 0, 1] \quad \text{(for GEMINI)}$$
$$\mathbf{B} = [0, 0, 0, 1, 0] \quad \text{(for CLAUDE)}$$

Now, applying the Euclidean distance formula:

$$d(\mathbf{A}, \mathbf{B}) = \sqrt{(1-0)^2 + (0-0)^2 + (0-0)^2 + (0-1)^2 + (1-0)^2}$$
$$d(\mathbf{A}, \mathbf{B}) = \sqrt{1^2 + 0^2 + 0^2 + (-1)^2 + 1^2}$$
$$d(\mathbf{A}, \mathbf{B}) = \sqrt{1 + 0 + 0 + 1 + 1}$$
$$d(\mathbf{A}, \mathbf{B}) = \sqrt{3}$$
$$d(\mathbf{A}, \mathbf{B}) \approx 1.732$$

Therefore, the Euclidean distance between the one-hot representations of GEMINI and CLAUDE is approximately 1.732.

7. Let $count(w, c)$ be the number of times the words w and c appear together in the corpus (i.e., occur within a window of few words around each other). Further, let $count(w)$ and $count(c)$ be the total number of times the word w and c appear in the corpus respectively and let N be the total number of words in the corpus. The PMI between w and c is then given by:

   (a) $\log \frac{count(w,c)*count(w)}{N*count(c)}$

   (b) $\log \frac{count(w,c)*count(c)}{N*count(w)}$

   (c) $\log \frac{count(w,c)*N}{count(w)*count(c)}$

   Correct Answer: (c)

   Solution: The correct answer is option (c): $\log \frac{count(w,c)*N}{count(w)*count(c)}$ Explanation: Pointwise Mutual Information (PMI) is a measure of association between two words, defined as the log of the ratio of their joint probability to the product of their individual probabilities.
   The formula for PMI is: $PMI(w, c) = \log \frac{P(w,c)}{P(w)P(c)}$
   Where:

   $P(w, c)$ is the probability of words w and c occurring together $P(w)$ is the probability of word w occurring $P(c)$ is the probability of word c occurring

   We can estimate these probabilities using counts:
   $P(w, c) \approx \frac{count(w,c)}{N}$ $P(w) \approx \frac{count(w)}{N}$ $P(c) \approx \frac{count(c)}{N}$
   Substituting these into the PMI formula:
   $PMI(w, c) = \log \frac{\frac{count(w,c)}{N}}{\frac{count(w)}{N} \cdot \frac{count(c)}{N}}$
   Simplifying: $PMI(w, c) = \log \frac{count(w,c) \cdot N}{count(w) \cdot count(c)}$
   This matches the expression in option (c), which is therefore the correct answer.

8. Consider a skip-gram model trained using hierarchical softmax for analyzing scientific literature. We observe that the word embeddings for 'Neuron' and 'Brain' are highly similar. Similarly, the embeddings for 'Synapse' and 'Brain' also show high similarity. Which of the following statements can be inferred?

   (a) 'Neuron' and 'Brain' frequently appear in similar contexts

   (b) The model's learned representations will indicate a high similarity between 'Neuron' and 'Synapse'

   (c) The model's learned representations will not show a high similarity between 'Neuron' and 'Synapse'

   (d) According to the model's learned representations, 'Neuron' and 'Brain' have a low cosine similarity

   Correct Answer: (a),(b)

   Solution: In skip-gram models, words appearing in similar contexts are given similar representations. Therefore, 'Neuron' and 'Synapse' will likely have high similarity in the learned representations.

9. Suppose we are learning the representations of words using Glove representations. If we observe that the cosine similarity between two representations v_i and v_j for words 'i' and 'j' is very high. which of the following statements is true?(parameter $b_i = 0.02$ and $b_j = 0.07$)

   (a) $X_{ij} = 0.04$

   (b) $X_{ij} = 0.17$

   (c) $X_{ij} = 0$

   (d) $X_{ij} = 0.95$

   Correct Answer: (d)
   Solution: GloVe representations mean:

   GloVe learns word vectors such that their dot product relates to the logarithm of words' co-occurrence probabilities X_{ij} represents the co-occurrence count between words i and j b_i and b_j are bias terms for words i and j

   Cosine similarity between vectors tells us:

   If it's very high, the words appear in very similar contexts This means they frequently co-occur with the same words In GloVe, this suggests they have high direct co-occurrence as well

   0.02: Too small for high similarity vectors
   0.2: Still relatively small
   0.95: High value consistent with high similarity
   0: Zero co-occurrence would not result in high similarity

   Therefore, option 3 ($X_{ij} = 0.95$) must be correct. This high co-occurrence value would explain the high cosine similarity between the word vectors, while being consistent with the given bias terms. The other values (0.04, 0.17, 0) are too small to result in vectors with very high cosine similarity, as they would indicate rare or no co-occurrence between the words.

10. Which of the following is an advantage of using the skip-gram method over the bag-of-words approach?

    (a) The skip-gram method is faster to train

    (b) The skip-gram method performs better on rare words

    (c) The bag-of-words approach is more accurate

    (d) The bag-of-words approach is better for short texts

    Correct Answer: (b)
    Solution: The skip-gram method performs better on rare words is the correct answer. Here's why:

    Skip-gram predicts context words given a target word It learns better representations for rare words because:

Each occurrence of a rare word gets multiple training examples (one for each context word) Each occurrence creates several context pairs This means more learning opportunities from limited data The model updates weights more times for each rare word appearance

a) The skip-gram method is faster to train

This is incorrect Skip-gram is actually typically slower to train than bag-of-words It generates multiple context pairs per word, increasing training time

c) The bag-of-words approach is more accurate

This is incorrect Neither approach is universally more accurate Each has different strengths and use cases Skip-gram often produces better quality word vectors

d) The bag-of-words approach is better for short texts

This is incorrect Bag-of-words can struggle with short texts due to sparsity Skip-gram can capture more meaningful relationships even in short texts